Weldezion, A. Y., Grange, M., Jantsch, A., Tenhunen, H., & Pamunuwa, D. (2015). Zero-Load Predictive Model for Performance Analysis in Deflection Routing NoCs. *Microprocessors and Microsystems*, *39*(8), 634-647. https://doi.org/10.1016/j.micpro.2015.09.002

Peer reviewed version

Link to published version (if available):
10.1016/j.micpro.2015.09.002

Zero-Load Predictive Model for Performance Analysis in Deflection Routing NoCs

Awet Yemane Weldezion[a,*], Matt Grange[b], Axel Jantsch[c], Hannu Tenhunen[a], Dinesh Pamunuwa[d]

[a] *The Royal Institute of Technology (KTH), SE 16440 Kista, Sweden*
[b] *Mentor Graphics, Oregon, USA*
[c] *Vienna University of Technology (TU Wien), Austria*
[d] *University of Bristol, UK*

Abstract

We study a static model for 2-D and 3-D networks that accurately represents the average distance travelled by packets under deflection routing, which is a specific form of adaptive routing. The model captures static properties of the network topology and the spatial distribution of traffic, but does not take into account traffic loading and congestion. Even though this static model cannot accurately predict packet latency under high load, we contend that it is a perfect predictor of deflection routing networks' relative performance under any load condition below saturation, and thus always correctly predicts the optimum network configuration. This is verified through cycle-accurate simulations of congested and uncongested networks with fully adaptive, deflection routing for regular traffic patterns such as uniform random, localized, bursty, and others, as well as irregular patterns in both regular and irregular networks. As the networks with minimal average distance perform best even under high traffic load, the average distance model establishes a robust relation between a static network property, average distance, and network performance under load, providing new insight into network behaviour and an opportunity to identify the optimal network configuration without time-consuming simulations.

Keywords: Alpha-model; average distance; b-model; hot-spot nodes; local traffic pattern; network optimisation; NoC; zero-load predictive model;

*Corresponding author
Email address: aywe@kth.se (Awet Yemane Weldezion)

1. Introduction

Analytical models of communication performance in networks are difficult to obtain because of the chaotic and complex nature of a congested communication system. The delicate balance between the switching, buffering, flow control, routing algorithm, and the traffic distribution across the network and over time determines whether a network operates at peak efficiency or exhibits overloaded and unbounded latencies. Predicting the expected packet delay in a network when it is near its saturation point is notoriously difficult. In fact, the specific load level that causes a network to become saturated not only depends on details of the spatial distribution (where packets are routed) and the burstiness (data injection patterns over time) of the traffic, but also on the history of the network's congestion.

To understand worst case timing, analytic models are indispensable and various methods have been applied to derive the worst case delay and performance in Networks-on-Chip (NoC) [1]. For instance, scheduling theory [2], network calculus [3], data flow analysis [4], and models used in statistical physics [5] are actively being pursued in the literature for networks that use deterministic routing. However, these models derive the upper latency bounds based on the worst possible interference patterns and congestion, which often is far from the average case. The task is even more daunting for adaptive routing. In deterministic routing networks, the delay of a packet only depends on direct and indirect interference on the packet's path. In contrast, adaptive routing balances the load over the entire network, which means that every packet may directly compete with any other packet. However, adaptive routing is a popular technique in NoCs due to its high performance [6, 7, 8], its load balancing capabilities [9, 10] and its fault-tolerant properties [11]. There have been attempts to exploit such properties by using a model-based approach in routing decisions [12], but there is no work known to us that offers an analytic delay model for average performance. Due to the exceedingly complex spatial and temporal interference patterns of packets across the network in adaptive routing networks, an accurate analytic latency model seems to be out of reach.

Consequently, simulation has been the predominant tool to assess the performance of networks for particular applications and application classes. The shortcomings of simulations are obvious: high effort in setting up realis-

tic simulations; even higher effort in setting up realistic application scenarios; very long simulation times; limited predictive value for application variants that are not simulated; and difficulties in obtaining clues for improving performance.

Given the challenges in formulating accurate analytic models and the enormous effort in setting up useful simulation scenarios, we ask the following question: Are there static, analytic properties that can serve as reliable predictors for network performance, even if their accuracy in predicting latency is limited?

In this work, we study one candidate for such a predictor: the average distance in hops that packets traverse in a network. In our model, a routing node consists of a router with one or more processing elements connected to it. Given any network topology, the geometric distance, expressed in the number of hops, between two specific routing nodes can easily be computed. For example, in a 3×4 2-D mesh network, the distance between any two neighbouring routing nodes is 1, and the distance between two diagonally opposite corner routing nodes is 5. If the probability of a specific node A sending a packet to a specific node B is known for all routing nodes A and B in the network, the average distance travelled by all packets for the given network topology and set of probabilities can be computed. We denote the average distance of a network by $\overline{\mathbb{H}}(\phi, \psi)$ (or $\overline{\mathbb{H}}$ for short), where ϕ is the spatial distribution of traffic and ψ represents the topology. We call this metric the *zero-load* average distance model as it models the average latency in networks completely free of congestion, or in other words networks with zero loading. We use the terms average-distance model and zero-load model interchangeably to mean the same thing.

We demonstrate the predictive power of $\overline{\mathbb{H}}$ by showing that for any topology with deflection routing, whether homogeneous or heterogeneous, under numerous realistic traffic scenarios, the model exhibits near perfect fidelity for all investigated cases. Fidelity is defined as the average latency for network 'A' being consistently less than network 'B' regardless of the congestion level and traffic pattern, when the average distance $\overline{\mathbb{H}}$ is less for network 'A' as predicted by the zero-load model. We examine the fidelity of our model by considering the packet latencies of networks that are equally sized in terms of total routing nodes, but have unequal radices (for example 4×4×4 versus 8×8×1 versus 2×4×8) as well as different configurations (different placements of specific traffic generators and consumers), under various traffic patterns with increasing injection rate. The zero-load model differentiates

between networks when other commonly used metrics, such as bisection channel bandwidth, B_c [13], can be inconclusive. For example, $B_c = 8$ for 8×8×1 and 2×4×8 meshes.

The main contribution of this work is to validate that the average distance model predicts relative network performance well for deflection routing networks, by means of a wide range of cycle accurate simulations using spatio-temporal traffic generators. Additional experiments are performed for placement of hot-spot nodes and IP-cores in irregular networks to demonstrate the potential of the model in network architecture optimisation.

The paper is organised as follows: in section 2, we discuss related works. In section 3 the different spatial traffic patterns are analysed and corresponding expressions for the zero-load average distance formulated. Also the basis for modelling bursty traffic is described. Section 4 describes the simulation environment and experimental methodology used in the study. In section 5 we validate our model by showing experimental results based on cycle-accurate simulations for regular network configurations under load for all the regular traffic patterns investigated.

Then, in section 6 we present results for irregular traffic patterns for both regular and irregular networks, and demonstrate the potential use of the model in optimizing network configuration. After discussing the results in 7, we draw our conclusions in section 8.

2. Related Works

The performance of communication networks has been widely studied and, in particular, there is a substantial body of work that deals with delay models for deterministic routing and regular topologies [1, 14, 15, 16]. Much less work has been done for adaptive routing networks, because the task is inherently more difficult. Therefore, all previous approaches make simplifying assumptions that make the task tractable but renders the model less general and restricts its scope.

One of the first delay models for adaptive routing networks was developed by Boura et al. in 1994 [17] for hypercube topologies. In 1998 Ould-Khaoua [18] reported a delay model for general k-ary n-cubes covering Duato's fully adaptive routing algorithm for wormhole switched networks and two or more virtual channels [19]. In 2000 Sarbazi-Azad et al. [20] proposed a modification which results in a model which has improved accuracy but is computationally very expensive because it recursively computes the

packet blocking delays in each node for every possible path a packet may take. In 2003 Khonsari et al. [21] provided an alternative delay model based on Boura's et al. earlier work [17] but for general k-ary n-cubes. It is less accurate but significantly faster to compute than the model of Sarbazi-Azad et al. [20].

These models assume a uniform spatial distribution and a Poisson process to model the temporal distribution of packet generation. In 2007 Min at al. [22] considered bursty traffic based on a compound Poisson process that models bursts, burst lengths and inter-arrival times of bursts as well as allowing exactly one hotspot. A model has also been proposed to predict the formation of hotspot traffic for the use of congestion-aware routing in certain networks [23].

All these delay models are fairly accurate only under the given assumptions, which are however, quite restrictive with regard to topology as well as traffic distribution. In relation to topology, some are restricted to hypercubes [17, 22], while all others target k-ary n-cubes [21, 20, 18]. No model available in the literature considers meshes (i.e. links do not wrap around peripheral nodes), which are popular for NoCs, or other regular or irregular topologies. All models except [22] assume and use Poisson processes for packet generation under a uniform spatial distribution. Min et al. do allow for bursty traffic and one hot-spot. However, self-similar traffic, found by many to closely resemble traffic flow in real applications [24, 25, 26], or spatial distribution of traffic beyond a single hot-spot, cannot be modelled. These are severe restrictions because real-world applications do not follow these idealistic assumptions. Relaxing or changing some of these assumptions requires a significant effort to adapt the delay model or develop a new approach without any guarantee of success. In contrast, our approach works for any topology and traffic pattern. We have collected evidence that it is valid and useful over a wide range of regular and irregular topologies and traffic patterns.

An even smaller number of works discuss the modelling and usage of the average distance as a performance metric. General zero-load latency models for different networks are described in [13, 27]. An approached based on average distance has been used by [28] to formulate models for static latency when accessing memory in large scale chip multiprocessors. In comparing network topologies, Agarwal [29] analysed the network latency for 2-D, 3-D and 4-D networks under localized traffic. The analysis is performed for zero-load and disregards the effect of congestion on the latency. It assumes that

routers and wires are the only constraints that affect delay. They report the following expression for the average distance in k-ary n-mesh networks:

$$\overline{\mathbb{H}} = \frac{n}{3}(k - \frac{1}{k}). \tag{1}$$

In practice, networks are rarely configured with equal radices. This is especially true with the advent of 3-D integration technologies. For a given network size the routing nodes are often arranged with different k_1, k_2 and k_3 radices. In formulating a simple adaptive partitioning strategy to minimise the communication cost, Liu et al. in [30] derived an expression for average distance in $k_1 \times k_2$ type 2-D mesh networks:

$$\overline{\mathbb{H}} = \frac{1}{3}(k_1 - \frac{1}{k_1}) + \frac{1}{3}(k_2 - \frac{1}{k_2}). \tag{2}$$

When $k_1 = k_2$, equation (2) is equivalent to Agarwal's equation (1).

In previous work [31] we showed how the average distance depends on the probability of transmission, $p_{i,j}$, of a packet with source i and destination j, and the actual source-destination Manhattan distance in terms of hops, with the following formulation for a 1-D network:

$$\overline{\mathbb{H}}_{1\times k} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} p_{i,j} \times |i-j|}{\sum_{i=1}^{k}\sum_{j=1}^{k} p_{i,j}}. \tag{3}$$

We went on to discuss unequal radices and formulated an average distance model for an n-D mesh that is the generalisation of Agarwal's and Liu's formulation:

$$\begin{aligned}\overline{\mathbb{H}}_{urt} &= \sum_{i=1}^{n} \overline{\mathbb{H}}_{urt_{1\times k_i}} \\ &= \frac{K_1}{3} - \frac{1}{3K_1} + \frac{K_2}{3} - \frac{1}{3K_2} + ... + \frac{K_n}{3} - \frac{1}{3K_n}\end{aligned} \tag{4}$$

Based on (3) we also derived average distance models for the spatial traffic patterns of uniform random traffic and local random traffic and verified the fidelity of the model by simulating networks under loading for these traffic patterns.

A more comprehensive approach is to use spatio-temporal traffic patterns that exhibit bursty characteristics, which is more representative of how real applications communicate over networks. Several studies have already shown that both system- and chip-level networks demonstrate properties of self-similarity [24, 26]. However, to our knowledge, no latency models have been published for spatio-temporal traffic patterns.

The network link bandwidth is dependent on the number of links available in the network. Depending on the configuration, the total number of links varies, even though the total number of routing nodes may be identical. This has been shown in [32] through comparative analysis of 2-D mesh and 3-D cube networks having the same routing node size. The expression for the total number of links is:

$$\begin{aligned} L_{2D} &= 4k_1k_2 - 2(k_1 + k_2) \\ L_{3D} &= 6k_1k_2k_3 - 2k_1k_2 - 2k_1k_3 - 2k_2k_3 \end{aligned} \tag{5}$$

which quantifies the differences in link bandwidth in different topologies.

Most analytical models take congestion into account to predict absolute network performance. The zero-load delay in such models can be deduced by setting the congestion level down to zero. However, finding zero-load delay in such an approach doesn't guarantee the accurate prediction of the relative performance of networks under load. Depending on the switching mechanism, and routing algorithm of the network, fidelity may not be maintained all the time. Also, the application of such models is limited to regular network topologies with regular traffic patterns.

In this paper, we broaden the scope of the study we presented in [31] by considering significantly more traffic pattern models including bursty and irregular traffic, as well as irregular networks. We evaluate the fidelity of the model in each case by comparing against results obtained from cycle accurate simulations, and demonstrate how it provides insight into the relative performance of differently configured networks under dynamic loading conditions. The study significantly expands on our previous work in terms of more experimentation and on understanding the underpinning theoretical concepts.

3. Traffic Patterns and Hop-count Models

Synthetic traffic models play an important role in design space exploration and verification. When an application runs, packets injected into the network

tend to exhibit repetitive spatio-temporal patterns that can be captured in a model [25]. The model should replicate both the temporal distribution, i.e. the timing of release of packets in the period under consideration, and the spatial distribution, i.e. the variation of destination addresses.

3.1. Spatial Distribution

Most works in the literature that propose synthetic traffic patterns discuss their spatial distribution, which determines how destination addresses are generated for packets. Spatial and temporal distributions are orthogonal to each other, and any temporal distribution can be superimposed on any spatial distribution. The set of destinations may contain only one node, resulting in a deterministic pattern, or it may include all nodes in the network with an associated probability. If the probability of transmitting to each node is identical, the traffic pattern is uniform random, while a probability that decreases with increasing distance results in localized traffic. In our experiments we utilise the following commonly used deterministic and probabilistic traffic patterns: uniform random (URT), bit reverse (BRT), bit complement (BCT), and local random (LRT) traffic.

For the purpose of defining spatial traffic patterns, routing nodes are assigned unique numbers $S = 0 \cdots N-1$ with N being the number of routing nodes. In a 3-D mesh topology the x, y and z address components are mapped from these routing node identifiers as follows:

$$\begin{aligned} x &= S \bmod N_x \\ y &= (S \operatorname{div} N_x) \mod N_y \\ z &= S \operatorname{div} (N_x N_y) \end{aligned} \tag{6}$$

where div is integer division and N_x, N_y, N_z denote the size of the network in each dimension. For a 2-D mesh the same equations hold except for the third, which becomes irrelevant.

For each traffic pattern, zero-load hop count models based on our original definition are expressly derived or stated below.

3.1.1. Uniform Random Traffic (URT)

In URT, the destination addresses are generated randomly and can be any processing element across the network other than the source itself [1]. For a

[1] Our convention does not allow a source to generate packets to itself. This does not detract from the generality, as simple modifications in the expressions can accommodate

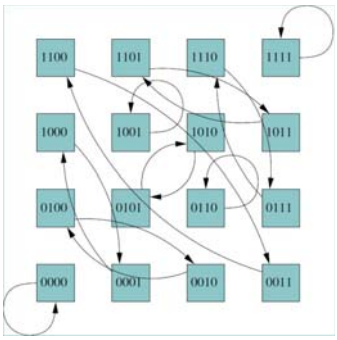


Figure 1: The bit-reverse traffic pattern with the node identifier shown in binary.

given network size of N routing nodes, URT creates a uniformly distributed spatial pattern, with equal destination probabilities for all source-destination pairs:

$$P_D = \frac{1}{N-1} \tag{7}$$

The overall average over the Manhattan distances associated with all source-destination pairs gives the average distance travelled by a packet across the network. In our previous work [31], the average distance expression in terms of hop count for a 3-D network under URT has been generalized as given in (8)

$$\overline{\mathbb{H}}_{urt} = \frac{1}{3}((x - \frac{1}{x}) + (y - \frac{1}{y}) + (z - \frac{1}{z})). \tag{8}$$

3.1.2. Bit-Reverse Traffic (BRT)

In BRT, the destination address is formed by reversing the binary format of the source node identifier as defined in section 3.1 and equation (6). For example, source node (001110) will send all its packets to destination node (011100). Figure 1 shows this pattern for a 4×4 network.

Let ς_n denote the bit-reverse of n, (e.g. $\varsigma_{100} = 001$), S the source node identifier, D the destination node identifier, and $S_x, S_y, S_z, D_x, D_y, D_z$ the address components of the source and destination nodes respectively [13].

the case with self-traffic. For example, equation (7) would have N instead of $N-1$.

Then equation (6) results in the following dependencies:

$$\begin{aligned} S_x &= S \bmod N_x \\ S_y &= (S \operatorname{div} N_x) \mod N_y \\ S_z &= S \operatorname{div} (N_x N_y) \\ D &= \varsigma_S \bmod N \\ D_x &= D \bmod N_x \\ D_y &= (D \operatorname{div} N_x) \mod N_y \\ D_z &= D \operatorname{div} (N_x N_y). \end{aligned} \tag{9}$$

If N is not a power of 2, i.e. $N \neq 2^k$, some bit-reversed values ς_S will be greater than N. Therefore we define $D = \varsigma_s \bmod N$. When $N = 2^k$, as in Figure 1, the modulo operation has no effect.

The distance between a source S and a destination D is the sum of the x, y and z differences:

$$\overline{\mathbb{H}}_{br,SD} = |D_x - S_x| + |D_y - S_y| + |D_z - S_z|. \tag{10}$$

For a network with N nodes, the average distance is expressed as the mean over all source-destination distances:

$$\overline{\mathbb{H}}_{br} = \frac{1}{N} \sum_{S=0}^{N-1} \sum_{D=0}^{N-1} \overline{\mathbb{H}}_{br,SD}. \tag{11}$$

3.1.3. Bit-Complement Traffic (BCT)

The destination node identifiers in the bit-complement pattern are derived by bit-wise complementing the source node identifier [13]. Figure 2 shows an example. If $\neg n$ denotes the bit-wise complement operation on a bit string n (e.g. $\neg 01011 = 10100$), then equation (6) gives:

$$\begin{aligned} S_x &= S \bmod N_x \\ S_y &= (S \operatorname{div} N_x) \mod N_y \\ S_z &= S \operatorname{div} (N_x N_y) \\ D &= \neg_S \bmod N \\ D_x &= D \bmod N_x \\ D_y &= (D \operatorname{div} N_x) \mod N_y \\ D_z &= D \operatorname{div} (N_x N_y). \end{aligned} \tag{12}$$

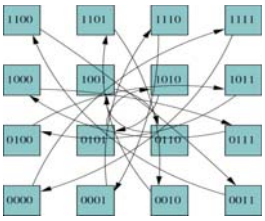


Figure 2: The bit-complement traffic pattern with the node identifier shown in binary.

As in the bit-reverse case, the distance between any source S and any destination D is

$$\overline{\mathbb{H}}_{bc,SD} = |D_x - S_x| + |D_y - S_y| + |D_z - S_z|, \tag{13}$$

giving the average distance for a 3-D mesh as:

$$\overline{\mathbb{H}}_{bc} = \frac{1}{N} \sum_{S=0}^{N-1} \sum_{D=0}^{N-1} \overline{\mathbb{H}}_{bc,SD}. \tag{14}$$

3.1.4. Localized Random Traffic (LRT) - the Alpha Model

In any architectural design, common sense dictates that components which communicate frequently with each other are placed in close proximity to avoid unnecessary delay and congestion, inasmuch as is possible within the physical constraints on placement. Local traffic models capture such sensible design decisions. Under a local traffic pattern, the probability of a given routing node being the destination for a generated packet varies inversely as the source-destination distance. Thus for any given source node, packets with close-by destinations are more numerous than packets with far-away destinations.

The level of localization can be explicitly specified in the model by the locality coefficient, α. When α=0, localization does not exist, and every node generates packets with equal probability to all nodes (always excluding self-traffic), whether near or far; this is identical to URT. As α increases, the localization effect increases and the number of packets generated with nearby destinations increases. As $\alpha \to \infty$, the average packet distance approaches 1 hop.

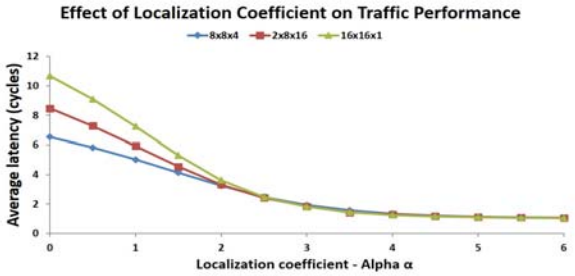


Figure 3: Effect of the variation of localization coefficient α on the average distance measured in hop counts. The hop count converges to 1 with increasing α.

For a given network size of N routing nodes the probability of sending a packet from S to D is

$$P_D = \frac{1}{K_S}.\frac{1}{|S-D|^{\alpha}} \tag{15}$$

for $S \neq D$, where $|S-D|$ is the geometric (Manhattan) distance and K_S is a normalizing factor that limits the sum of all probabilities to 1. Its value is different for each source S and is calculated as follows:

$$K_S = \sum_{D=0}^{N-1} \frac{1}{|S-D|^{\alpha}}. \tag{16}$$

Then the average hop count is derived as follows:

$$\overline{\mathbb{H}}_{\alpha,SD} = \frac{1}{N-1} \sum_{S=0}^{N-1} \sum_{D=0}^{N-1} (|S-D|P_D). \tag{17}$$

Substituting (15) in (17) results in:

$$\overline{\mathbb{H}}_{\alpha,SD} = \frac{1}{N-1} \sum_{S=0}^{N-1} \sum_{D=0}^{N-1} \frac{|S-D|^{1-\alpha}}{K_s}. \tag{18}$$

The localization effect varies according to the network size and topology. Figure 3 shows the localization effect on the average distance for a network of 216 routing nodes arranged as 8×8×8 and 2×8×16 cuboids and a 16×16×1 mesh. When α=0, the average hop count is the same as with URT, though the values are different for each configuration. As α increases, the localization of traffic increases, and the average distance decreases until all curves converge to a value of 1 hop count.

3.2. Temporal Distribution

The temporal distribution defines the timing of release of packets into the network. Several studies have concluded that realistic network traffic demonstrate the property of *self-similarity* over a long period of time [24, 25, 26]. As bursty traffic is very prevalent in real applications, we have established a self-similar synthetic pattern as a bursty traffic model which emulates realistic streaming of data.

Discrete self-similar traffic can be modelled by the bursty model (B-Model) as described in [25]. In the B-Model, a bias β ($0 < \beta < 1$) is introduced to the streaming pattern. A bias β=0.5 indicates that packets are streamed at a uniformly distributed rate throughout a time interval comprising, say, n cycles. When the bias is set below or above 0.5, the streaming rate becomes skewed, with the n-cycle time interval being split into two equal portions, and a specified fraction of packets being emitted in the first half, and the rest in the second half. For example, a bias of β=0.2 implies that 20% of the packets are streamed in the first half and 80% in the second half of the time interval under consideration, or vice versa. This process of halving is continued for each generated half of the original interval, for a number of times that is defined as the depth d, resulting in some number of discrete time intervals in which the packets are distributed. For an n-cycle time series, the number of such discrete intervals in the final sequence is given by $\frac{n}{2^d}$. The maximum value for d is limited by the inequality $\frac{n}{2^d} \geq 1$ or $d \leq \log_2(n)$ (where the simulation cycle duration has been normalised to 1), as a simulation cycle is an indivisible, atomic unit of time. After each division, the choice of which half is assigned 20%, and which 80% (in this example), is made randomly.

The total number of packets, ϕ_{total}, to be transmitted within the time-series of n cycles depends on the injection rate, γ, ($0 \leq \gamma \leq 1$). At the maximum injection rate ($\gamma_{max} = 1$) a node can inject at most one packet per clock cycle, i.e. n packets in an n-cycle time-series. In general the following relationship holds:

$$\phi_{total}(\gamma) = \gamma n. \tag{19}$$

The physical time at the beginning of period i in the sequence is given by $i\frac{n}{2^d}$. If the total number of packets allocated to each period in the final sequence is $x(i\frac{n}{2^d})$, the total traffic volume (total number of packets) is the

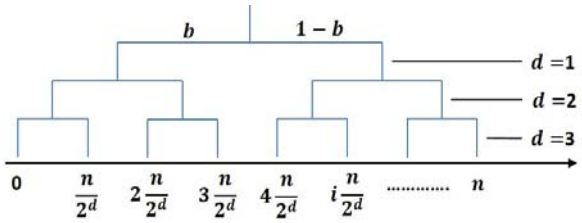


Figure 4: Allocation of packets in the B-model. The original n-cycle time-series is divided into a number of discrete time intervals by a process of continuously splitting the parent sequence into two halves. In the first step, the sequence comprises the entire time series, and hence there is one interval which contains all of the packets. Halving this interval for a number of times, d (depth), and randomly allocating to each resulting interval a fraction of either β or $1-\beta$ of the total number of packets in the parent sequence results in the final distribution of packets over time.

sum of the traffic volume in each period:

$$\phi_{total}(\gamma) = \sum_{i=0}^{2^d} x(i\frac{n}{2^d}). \tag{20}$$

The number of packets that a node injects into the network within any period, $x(i\frac{n}{2^d})$, can be expressed as a function of the bias, β, the division depth, d, and the injection rate, γ

$$x(i\frac{n}{2^d}) = (\{\beta, 1-\beta\})^d(\gamma n - \sum_{j=0}^{i-1} x(j\frac{n}{2^d}). \tag{21}$$

In (7), the traffic volume at a given point in the final time sequence is defined as a function of the traffic volume at the coarser time step, and has a straightforward recursive implementation.

Figure 5 shows the distribution of 1,000 packets over 10,000 cycles with a bias of $\beta = 0.2$ and an injection rate of $\gamma = 0.1$. If the total is increased to 2,000 packets ($\gamma = 0.2$), the only change is in the amplitude (y-axis). The temporal distribution (x-axis) is identical.

It turns out that the temporal distribution of packet generation has no impact on the average hop count. The distance is determined by locations of source and destination of packets, but not when in time they travel. Consequently, it has no effect on the average hop count metric. However, it is well known that bursty traffic is unhealthy for networks. Given a certain amount

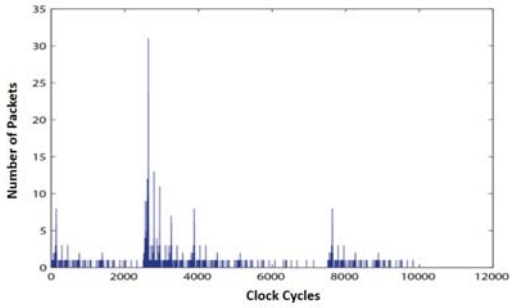


Figure 5: Distribution of 1000 packets over 10,000 cycles according to the B-model with β=0.2.

of traffic to be transmitted, networks handle smooth traffic flows much better than bursty traffic with big spikes. Viewed differently, a network needs more buffering resources to cope well with bursty traffic.

In this light one can suspect that the average distance model has difficulties to predict the relative performance of networks if the traffic is very unevenly distributed over time. Intuition would suggest that two network configurations may have significantly different capabilities to handle bursts, even if both of them exhibit the same average distance. It therefore came as a surprise to us that this effect did not appear in any of our simulations, as exemplified by Figures (7, 8, 10, 11, 13, 14, 16, 17). Certainly, more bursty traffic results in heavier loading and causes the network to saturate at a lower injection rate. However, this affects all alternative network configurations in the same way and any differences of networks to cope with bursts are evened out by the load distribution capability of the deflection routing algorithm.

4. Simulation Environment and Comparison Methodology

In this section we describe the simulation environment and experimental setup. In our network simulator, a hop is counted when a packet traverses the link between adjacent routers.

4.1. Traffic configuration

The average distance predicted by the zero-load model for the various traffic patterns as described in section 3 is calculated numerically for three network topologies having the same total number of routing nodes, 4×4×4,

2×4×8 and 8×8×1. Each spatial traffic model is then combined with a self-similar temporal bursty traffic model with bias values of 0.1, 0.3 and 0.5, and used to generate traffic for cycle-accurate register-transfer-level (RTL) simulations. The zero-load case is emulated by having a very low injection rate of 0.01 packets per node per cycle, and the fidelity of the model in predicting the network performance is checked by increasing the injection rate beyond the saturation point. For a range of injection rates within the simulation period, the average latency values are calculated for packets collected from a sample window defined within the stable phase of the network (after the warm-up phase and before the cool-down phase). For the specific router micro-architecture considered, a single hop count is equivalent to five clock cycles in simulation. The simulation window is always long enough that no packets are dropped.

4.2. On-Chip Network Architecture

A hop count can be translated into network latency given that the physical constraints are known. Ideally, the router-to-router hop delay is equal throughout the network. This assumes that the link sizes are the same and that all routers are identical. If the network is not regular, router-to-router length is not the same and hops cannot be directly converted to network latency. An example is a 3-D cube network where through silicon vias (TSVs) are used to connect the vertical layers. The TSVs are short and fast compared to long global planar wires due to their lower electrical parasitics. As a result, inter-layer communication is typically faster than intra-layer communication. This means that horizontal networks hops are slower than vertical hops. Thus, vertical and horizontal hops are separated to calculate the network latency.

In this study we use a buffer-less switch and non-minimal, fully adaptive, deflective routing, also known as hot-potato routing. Buffer-less routers have an inherent advantage of simplicity, energy-efficiency, and cost-effectiveness [33]. Different implementations of buffer-less architectures have been reported [34] [35]. Each router consists of control units and sorting units and utilise a crossbar architecture, pipelined in three stages, with connectivity between input/output ports to six directions and to the resource. The six directional ports are North, South, East, West, Up, and Down with the seventh port providing access to the resource (such as on-chip processing elements or off-chip blocks such as memory or I/O).

Mesh networks are used throughout to connect routers in 2-D, and cube networks in 3-D. Meshes are chosen because of their simplicity in configuration and practicality in hardware implementation. Depending on its position, a single router connects between two and four routers in its Manhattan neighbourhood in a 2-D mesh; a router connects between three and six routers in 3-D cube networks.

A packet is a single flit long containing both control and payload bits. Once the destination address is provided by the source, the packetization process is initiated in a network interface (NI) component. A relative addressing scheme is used to set the destination bits in the form of X, Y, Z. For temporal traffic, a self-similar pattern is used; the spatial traffic patterns comprise variously uniform, bit-reverse, bit-complement, hot-spot, and local patterns. When running simulations the injection rate is varied depending on the traffic pattern in use.

When two or more packets compete for the same link, we honour an oldest packet first priority scheme. No packets are dropped from the network. Instead, when the network is congested, the packets are accumulated in a FIFO buffer in the network interface (NI) situated between each router and its local processing element. More details of the routing protocol and router micro-architecture are given in [36].

4.3. Simulation Methodology

Packet latencies are extracted by running cycle accurate RTL simulations, collating packet injection, ejection and traversal data at each router over the entire simulation and processing this data in Matlab. The latency in all graphs is given in multiples of clock cycles. This allows a straightforward comparison between regular and irregular networks where a hop comprises different numbers of clock cycles. More details of the simulation methodology are given in [36].

5. Experiments on Regular Traffic Patterns

In this section and the next, we show how the zero-load predictive model exhibits almost perfect fidelity for regular and irregular traffic patterns and network topologies, and any deviation is within the limits of numerical accuracy imposed by the simulations and calculations. We also show how the model can be used to find the optimum network node placement within the limits allowed by the communication and physical constraints imposed by

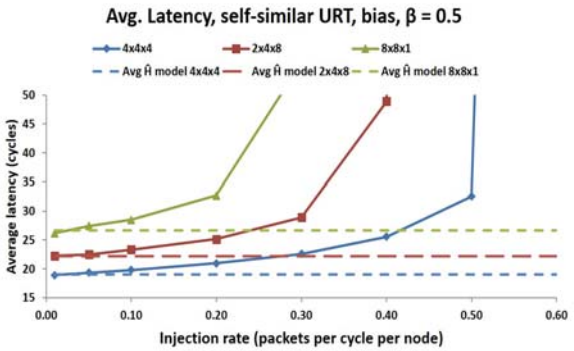


Figure 6: Variation of average latency with increasing injection rate for non-bursty URT with bias, β=0.5

the specifications. This section concentrates on *regular traffic patterns* and *regular networks*, characterised by homogeneity across the network in both cases. Section 6 looks at *irregular traffic patterns* and *irregular networks* characterised by heterogeneity across the network.

5.1. Uniform Random Traffic (URT)

Figure 6 plots the simulation results for a 64 routing node network configured as an 8×8×1 2-D mesh, and 2×4×8 and 4×4×4 3-D meshes. Packets are injected under the URT model with no burstiness (i.e. bias β=0.5). At very low injection rates, the average hop counts are equal to the zero-load delay in terms of clock cycles. The configuration with the minimum average distance is the $4 \times 4 \times 4$ 3-D mesh, as its geometry dictates that packets have to traverse fewer links to reach their destinations. When the injection rate is increased the network congestion levels increase, and as a consequence the average delay grows for all configurations.

Interestingly, increasing injection rates *increase* the differences between these configurations under load. We observe this phenomenon in many, but not all traffic patterns, with hot-spot being a notable exception as discussed in section 6.1.

Figures 7 and 8 show the growth of latency with increasing injection rate for bursty URT with bias β=0.3 and β=0.1 respectively. While the saturation injection rate drops, the zero-load average distance model exhibits perfect fidelity.

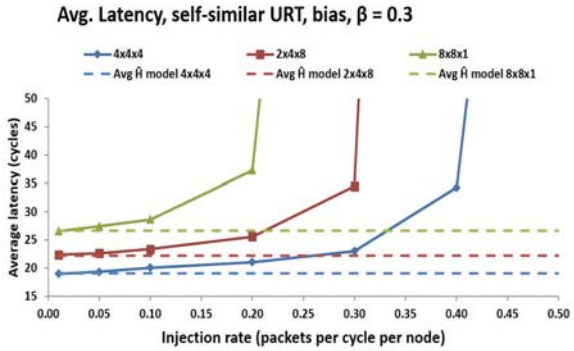


Figure 7: Variation of average latency with increasing injection rate for bursty URT with bias β=0.3.

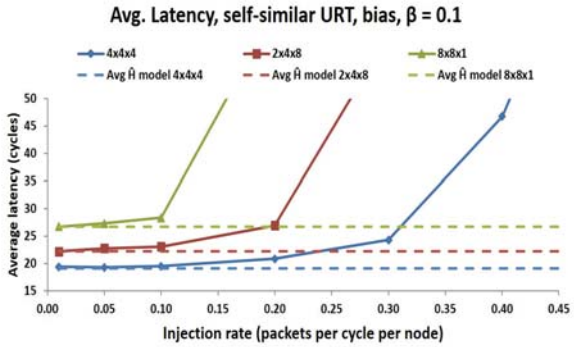


Figure 8: Variation of average latency with increasing injection rate for bursty URT with bias β=0.1.

5.2. Bit-Reverse Traffic (BRT)

The injection rate is varied from 0.01 up to 1.0 packets per node per cycle for biases of β=0.5, β=0.3 and β=0.1 under the self-similar temporal model for bit-reverse traffic. Figure 9 shows the result for β=0.5 which is equivalent to the case of packets being uniformly distributed in time according to the bit-reverse spatial pattern. When the bias is skewed to β=0.3, as shown in Figure 10, the average distances start to increase in each case due to the increased congestion in the network and the network exit points. This worsens when the bias is set to β=0.1, as shown in Figure 11. In all cases, the zero-load model predicts the relative performance of the configurations correctly up to the saturation point.

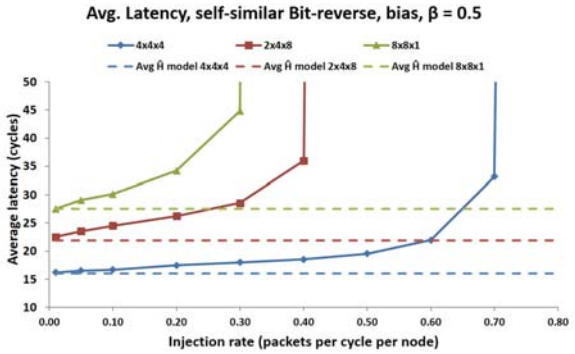


Figure 9: Variation of average latency with increasing injection rate for non-bursty BRT with bias, β=0.5.

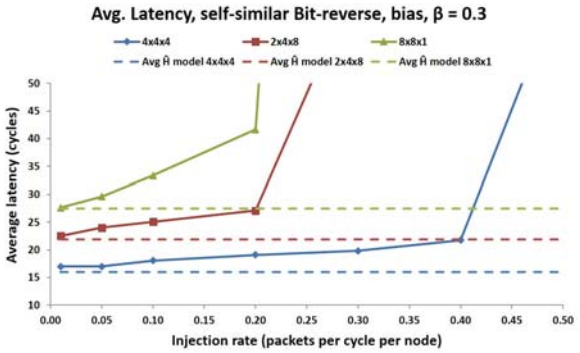


Figure 10: Variation of average latency with increasing injection rate for bursty BRT with bias, β=0.3.

5.3. Bit-Complement Traffic (BCT)

Figure 12 shows the results for unbiased traffic with the bit-complement spatial distribution. For low injection rates the average latency converges to the delay predicted by the zero-load model in terms of clock cycles for each configuration. When the injection rates are increased, the latency also increases without the curves ever crossing each other. Similarly, we observe also perfect fidelity for bursty traffic with bias β=0.3 and β=0.1 as shown in Figures 13 and 14 respectively.

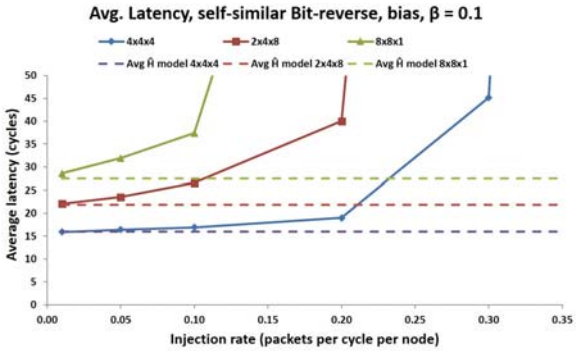


Figure 11: Variation of average latency with increasing injection rate for bursty BRT with bias, β=0.1.

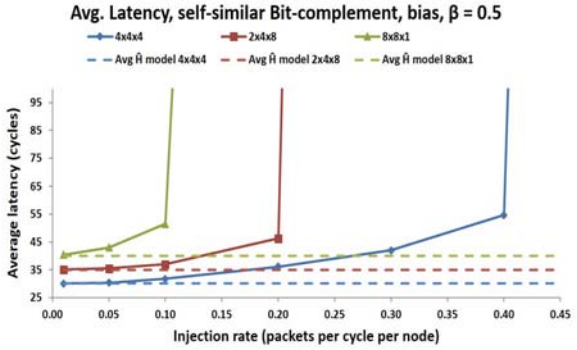


Figure 12: Variation of average latency with injection rate for non-bursty BCT when bias, β=0.5.

5.4. Localized Random Traffic (LRT)

Figure 15 shows how the latency increases with injection rate when the localization coefficient α=1 and the self-similar bias β=0.5, ensuring uniform streaming of packets under a local traffic pattern. At low injection rates the latency converges to the zero-load average distance as for the other cases. When the bias is set to β=0.3 (Figure 16), or β=0.1 (Figure 17), the resulting temporally skewed traffic causes insignificant changes. This is because strong localization in the traffic generation results in more packets with destinations within a relatively short distance compared to the network dimensions. Clearly, for each configuration, the average latency for local traffic is less than the corresponding URT traffic shown in Figures 6, 7 and 8.

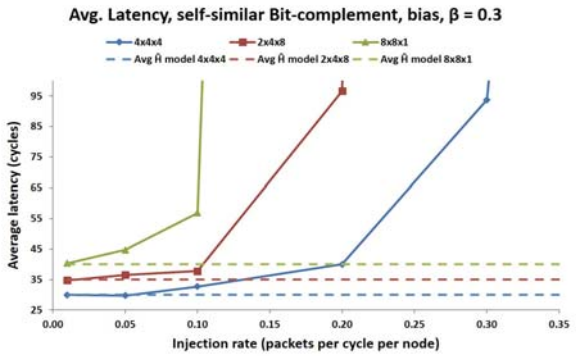


Figure 13: Variation of average latency with increasing injection rate for bursty BCT with bias, β=0.3.

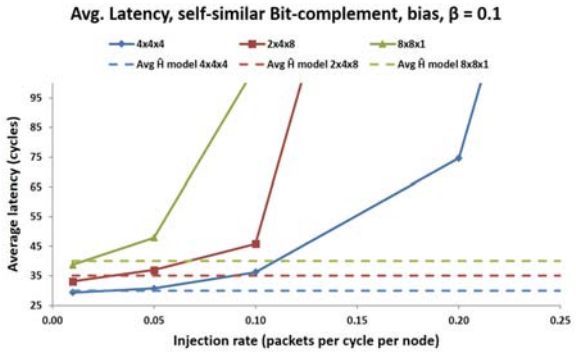


Figure 14: Variation of average latency with increasing injection rate for bursty BCT with bias, β=0.1.

6. Experiments on Irregular Traffic Patterns

In this section, we further validate the zero-load predictive model for networks with irregular traffic patterns as well as irregular networks. We also show how such networks can be configured for optimal performance.

6.1. Networks with Hot-Spots

Nodes that generate or receive a greater proportion of traffic than other nodes are called hot-spots. Typical hot-spot nodes are memory controllers, a critical processing resource, or a system controller.

For instance, the wide-IO JEDEC standard specifies 512 bit wide data interfaces [37] from the logic plane to the DRAM memory plane in stacked

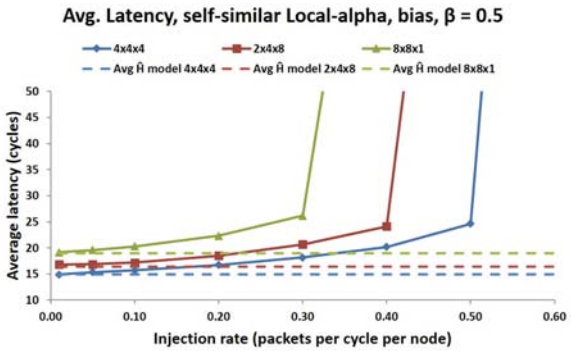


Figure 15: Variation of average latency for non-bursty (bias, β=0.5) LRT with alpha α=1.

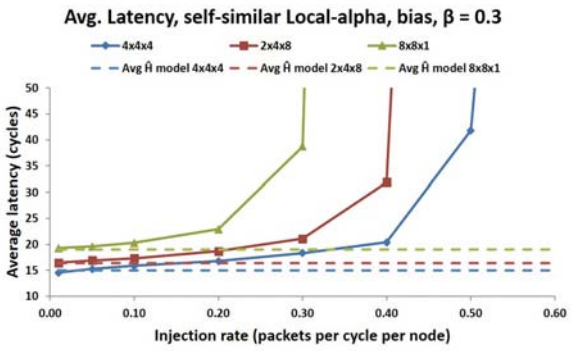


Figure 16: Variation of average latency for LRT with α=1 and bias β=0.3.

.

systems. DRAM layers can be physically stacked on top of (or below) logic layers and connected by means of through-silicon vias (TSVs). Each wide I/O access port requires 512 interconnects for data and additional lines for addressing. The processing elements or cores in the logic layers typically share the memory layers. This means that data access is made only through the parallel TSV clusters, which in turn are accessed on the die through a dedicated resource. Such shared access creates a hot-spot region in an on-chip network architecture. Hot-spot regions should be designed in such a way that there is sufficient link bandwidth to support worst-case traffic congestion. This leads us to explore the optimal placement of hot-spot nodes on a die to minimise congestion, given placement constraints.

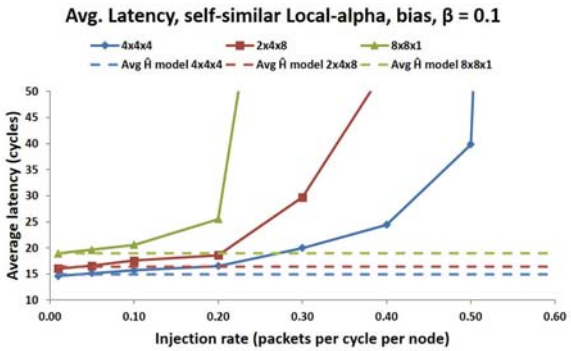


Figure 17: Variation of average latency for LRT with $\alpha = 1$ and bias $\beta = 0.1$.

.

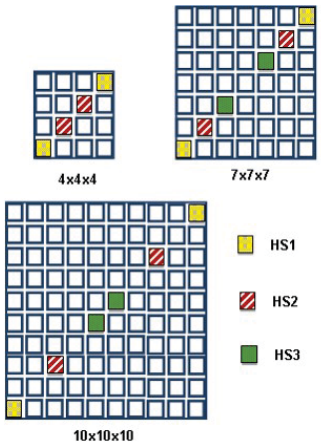


Figure 18: Placement of hot-spot nodes on top layer.

Figure 18 shows different configurations of two resources serving as access ports to DRAM either stacked in the same package or placed off-chip. The memory access resources have to be on the top logic layer due to I/O considerations. Each core in the network in any of the three layers that has access to any block in the memory layers sends requests through the access ports. The combined requests generate a hot-spot region with heavier traffic in the area surrounding these tiles.

The optimal placement of these hot-spots that yields the best performance is found through cycle accurate RTL simulations for networks under loading. For this experiment, we examine networks of three different sizes, 4×4×4,

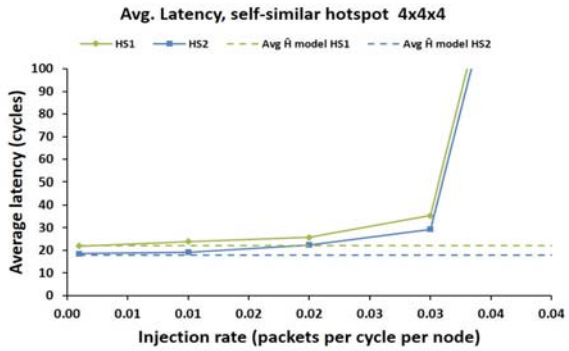


Figure 19: Variation of average latency with injection rate for hot-spot traffic in 4×4×4 network with HS1 and HS2 hot-spot placement on top layer

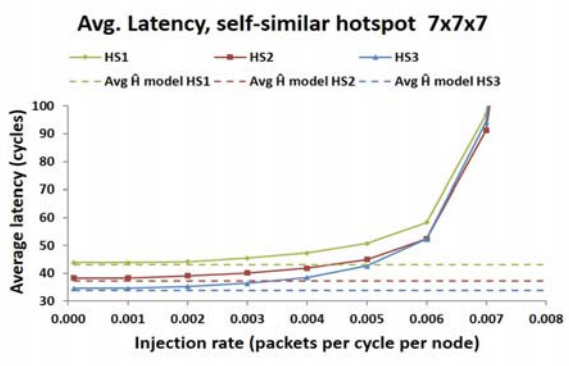


Figure 20: Variation of average latency with injection rate for hot-spot traffic in 7×7×7 network with HS1, HS2 and HS3 hot-spot placement

7×7×7, and 10×10×10. The different placements of the access tiles on the top layer considered are shown in Figure 18. While some other arrangements are possible, many can be eliminated through symmetry, and these are carefully selected as being representative of most sensible configurations to validate the case.

In this exercise hot-spot nodes receive 80% of the packets generated by the non-hot-spot nodes, while the remaining 20% are sent to other non-hot-spot destinations under a uniform random distribution. This spatial distribution is then uniformly distributed over time (bias β=0.5 in the self-similar model). Figures 19 to 21 show the results for each network configuration with increasing injection rate from 0.001 up to 0.01 packets per node per

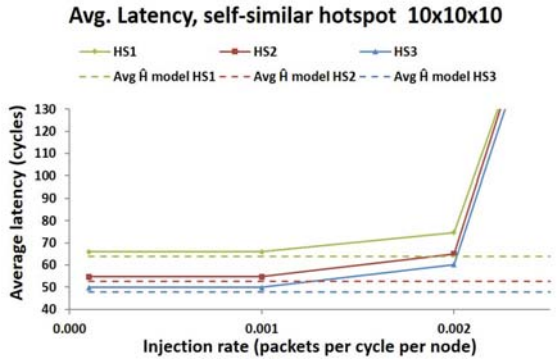


Figure 21: Variation of average latency with injection rate for hot-spot traffic in 10×10×10 network with HS1, HS2 and HS3 hot-spot placement

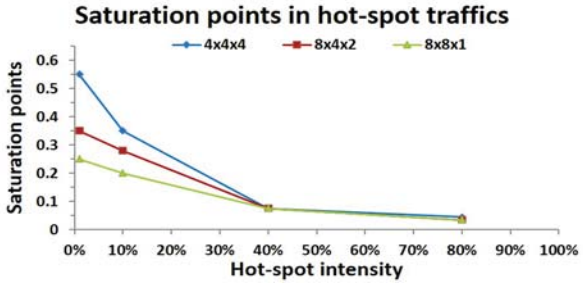


Figure 22: The three configurations have an equal no. of routing nodes and each has two hot-spots located at their center. The Y-axis gives the saturation injection rate while the X-axis denotes the fraction of overall traffic directed to the hot-spots. As this fraction increases, the networks' saturation injection rates decrease and converge.

cycle.

With increasing injection rate, the average packet latency in each configuration increases without the curves crossing each other. The model again exhibits perfect fidelity for all tested hot-spot configurations.

It is interesting to note that the differences in latency of different configurations decrease as the network load increases, unlike all the traffic patterns studied earlier. For URT, LRT, BRT and BCT the differences grow because the longer a packet has to travel the more it will suffer from increased congestion simply because there is more time for it to be affected. As the less optimal topologies have on average more packets that travel longer, they are affected more by congestion and hence the differences in latency increase.

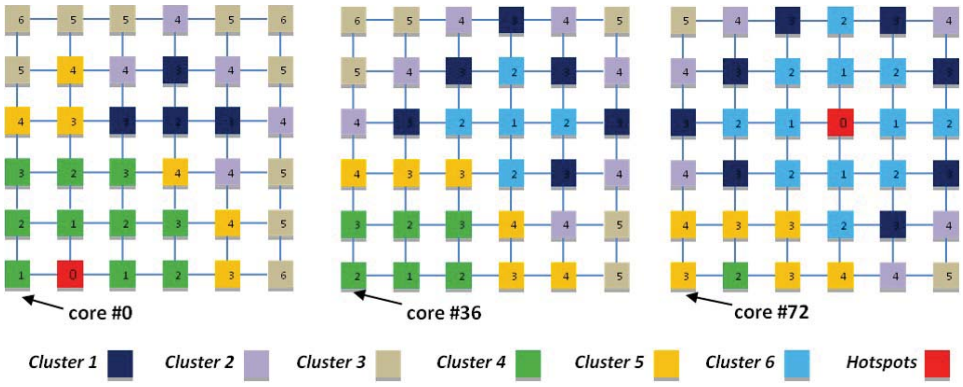


Figure 23: A 6×6 network configuration in 3 layers with node clustering

	M451	M549	M470	M452	M450	M245
Model	4.6041	4.6062	4.6085	4.6122	4.6451	4.888
d(M451)	0.00%	-0.05%	-0.10%	-0.18%	-0.88%	-5.81%
d(M549)	0.05%	0.00%	-0.05%	-0.13%	-0.84%	-5.77%
d(M470)	0.10%	0.05%	0.00%	-0.08%	-0.79%	-5.72%
d(M452)	0.18%	0.13%	0.08%	0.00%	-0.71%	-5.64%
d(M450)	0.89%	0.84%	0.79%	0.71%	0.00%	-4.97%
d(M245)	6.17%	6.12%	6.06%	5.98%	5.23%	0.00%

No Cross-over
Cross-over
Self-reference

Figure 24: Percent differences for selected configurations

With hot-spot traffic, packets that travel to a hot-spot or nearby a hot-spot will suffer more than other packets from congestion. As load increases, the congestion around hot-spots will rise first, affecting all packets that travel near-by indiscriminately. Therefore we see in Figures 19, 20, and 21 that the latency curves of the different configurations increase roughly in parallel until the congestion starts to dominate the delay, at which point the lines converge. This convergence of saturation points is demonstrated in Figure 22. It shows three network configurations with two hot-spots around their respective centres. As the fraction of traffic directed to these hot-spots increases, the injection rate at which these different configurations saturate decrease and converge. Hence, strong hot-spot traffic tends to dominate a loaded network, defines its saturation point, but does not, even under high load, reverse the relative performance ranking of networks. Consequently, the average distance model is a valid predictor for hot-spot dominated networks. See section 7 for a discussion of when this trend is likely to be reversed.

Table 1: Traffic probabilities for MAP IP-cores

Source IP cores	Probability to target IP-core	Relative Injection rate
GPU	68% L2 GPU, 2% CPU, 20% Display Interface, 9% total to all other interfaces, 1% System control	1 IR
CPU	40% L2 CPU, 8% All GPU, 10% Audio, 10% Video, 4% Camera, 5% Security 22% all other Interface, 1% System control	0.7 IR
Audio	30% WideIO, 28% Security, 20% CPU, 15% Standard, 3% Ethernet, 3% User, 1% System control	0.2 IR
Video	50% WideIO, 9% Security, 20% CPU, 20% all interfaces, 1% System control	0.8 IR
Camera	30% WideIO, 60% Display, 5% CPU, 4% Security, 1% System control	0.8 IR
Security	60% WideIO, 20% Audio, 14% Video, 5% CPU, 1% System control	0.3 IR
L2 GPU	19% L3, 80% GPU, 1% System control	0.8 IR
L3	26% L2 GPU A, 26% L2 GPU B, 26% L2 CPU, 21% WideIO, 1% System control	1 IR
L2 CPU	20% L3, 79% CPU, 1% System control	0.8 IR
WideIO	48% L3, 15% Audio, 26% Video, 5% Security, 5% Camera, 1% System control	1 IR
System Control	24% CPU, 24% GPU, 4% To every remaining 13 cores,	0.2 IR
Standard Interface	16% GPU, 20% CPU, 46% Audio, 10% Video, 5% Security, 2% Camera, 1% System control	0.5 IR
User	24% GPU, 22% CPU, 24% Audio, 24% Video, 5% Security, 1% System control	0.5 IR
Ethernet	24% GPU, 25% CPU, 15% Audio, 30% Video, 5% Security 1% System control	0.5 IR
Display	64% GPU, 12% Video, 15% CPU, 12% WideIO, 3% Camera, 5% Security,1% System control	0.5 IR

6.2. Configuration of Regular Networks with Irregular Traffic Patterns

In this example we attempt to identify the best placement configuration in a complex 3-D network by means of the zero-load predictive model. The network has three layers with each layer having 6×6 nodes as shown in Figure 23. The network includes two hot-spot nodes. The first provides access to off-chip data inputs and outputs, and is placed at the periphery of the bottom layer based on I/O considerations. The second provides access to a wide-IO port that connects to a memory layer stacked on top of the three layers. It is placed in the middle of the network based on manufacturing considerations [38]. In order to simplify the traffic allocation, the routing nodes are grouped into six different clusters defined by their traffic generation probability as shown in table 2.

Table 2: Traffic generation probabilities of cores in different clusters to hot-spot nodes (memory & off-chip) and other nodes

Number of cores	To Memory	To Off-chip	To other cores
Cluster 1=18	7.14%	7.14%	85.71%
Cluster 2=18	14.29%	14.29%	71.43%
Cluster 3=18	21.43%	21.43%	57.14%
Cluster 4=18	28.57%	28.57%	42.86%
Cluster 5=18	35.71%	35.71%	28.57%
Cluster 6=18	42.86%	42.86%	14.29%

By permutation of the six clusters in the network, 720 possible configurations (M001-M720) can be derived. The zero-load average distance model reveals that configuration M451 with 4.6041 hops has the shortest average distance whereas M245 with 4.888 hops has the longest average distance. Figure 24

shows the first five top configurations (M451, M549, M470, M452, & M450) with shortest average distance as well as the one with the longest (M245) in the top row. The relative % difference between any two configurations is calculated.

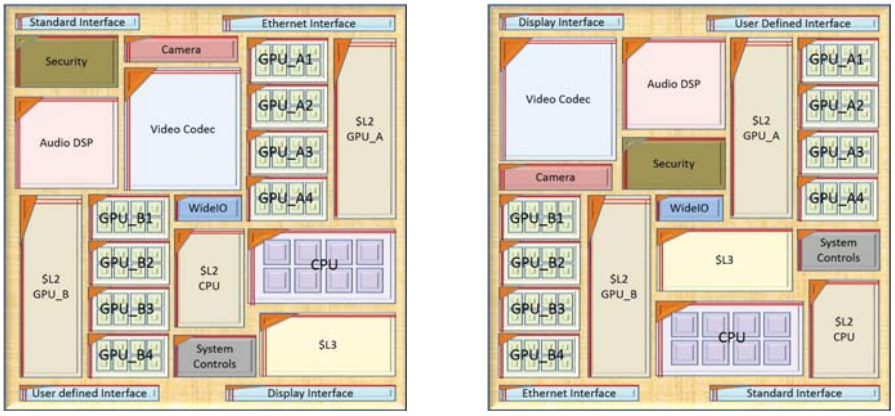
The difference in average distance between all six configurations are also shown in Figure 24. For example, the relative difference of M245 and M451 is 6.17%. Given the large number of possible configurations, the difference between any two consecutive configurations is quite small.

In order to check the fidelity of the model and see how the predictions hold up under increasing load, we carried out RTL simulations for all configurations.

In this example we did find cross-overs in the latency curves for different configuration, which are marked as red cells in Figure 24. It turns out that cross-overs only occurred when the difference in the zero-load average distance was less than or equal to 0.13%, which translates to an absolute difference on the order of 0.006 for the zero-load average distance values prevalent in this example. In the post-processing, latency values are rounded to two decimal places and truncated, leading to a maximum absolute error in individual readings of 0.005, which can accumulate over multiple transactions. Also, the stochastic processes used to generate packets over time for the simulations deviate in any finite time period from the ideal probability distributions used in the zero-load model. Therefore these cross-overs seem to be within the range of the numerical error introduced by the simulations and proceeding calculations and appear not to represent a true violation of the fidelity of the model.

6.3. Configuration of Irregular Networks with Irregular Traffic Patterns

The relevance of the zero-load predictive model for different applications is further investigated with an irregular network, based on two configurations of a generic mobile application processor (MAP) shown in Figure 25(a) and 25(b). The MAP is composed of processing elements of different sizes each connected to a routing node and thus the network topology used to connect them is an irregular one. We have used the following core descriptions to obtain likely tile sizes for the network as well as representative (irregular) traffic pattern models of communication between the different elements. The MAP consists of GPU clusters each with four GPU cores, and a single CPU cluster with eight cores. Each cluster accesses its own dedicated L2 cache. There is a common L3 cache with direct access to a 3-D wide-IO port located



(a) MAP Model 1 (b) MAP Model 2

Figure 25: Two configurations of a generic Mobile Application Processor (MAP)

at the center. The wide-IO DRAM blocks are stacked on top of the MAP. There are also application specific IP-cores such as an audio DSP and video codec, a camera, and security and system controls. For off-chip accesses, a standard interface such as USB, SPI or any user defined interface can be used. A display interface and wired connectivity through Ethernet is also included.

The traffic generated by individual IP-cores is non-uniform. Cores such as GPUs stream packets at a higher rate while IP-cores such as system controllers generate packets at a lower rate with a small contribution to the overall traffic. Table 1 shows the spatial probability distribution of traffic used to simulate the two MAP configurations, normalised to the GPU injection rate. For example, the relative injection rate of 0.2 for the security IP-core means that its traffic contribution is only 20% of the maximum traffic contribution by a core (i.e. the GPU's contribution). The simulation results are shown in Figure 26, and the results confirm 100% fidelity of the model for injection rates below saturation.

7. Discussion

In the absence of reliable delay models for NoCs with adaptive routing, cycle-accurate simulation is the only tool to assist system architects in deciding upon network topology, mapping, and other critical early-phase design

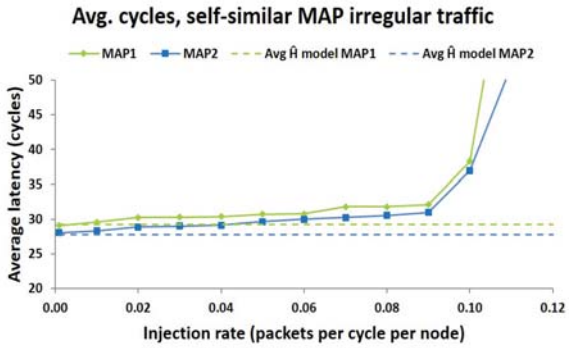


Figure 26: Average clock cycles with self-similar irregular traffic pattern for MAP configurations

choices. By focusing on a *relative* rather than absolute performance metric, we have formulated a model that predicts with high fidelity whether one configuration will exhibit better performance than another even under high load with burstiness in packet injection. The zero-load model is a static property of topology, mapping and traffic probabilities. Even though it does not take into account congestion, interference or temporal variability in traffic, it surprisingly shows almost perfect fidelity for deflection routing networks. We studied the model under a wide range of loading and topological conditions from uniform random to hot spots to irregular traffic and networks. Under all these conditions we only observed the relative performance of different network configurations changing under load in a few cases when the average distance of two alternative configurations differed by 0.13% or less. These differences fall within the numerical error introduced by rounding and stochastic variations in the traffic generation.

It is interesting to note that in all studied cases of regular traffic patterns the differences in delay grow with increasing traffic load, as attested to by the diverging delay curves in Figures 6-17. For hot-spot and other irregular traffic patterns the curves run parallel (Figures 19, 21, 26) or even converge (Figure 20). It seems that divergence occurs when congestion builds up uniformly in the whole network, thus aggravating every initial difference. However, if network behaviour is dominated by the congestion in a small area, the saturation point is reached when this small area becomes heavily congested, and thus any initial advantage in terms of the average distance is lost. Thus, in these scenarios the delay curves converge towards the same

saturation point (Figure 22). A prime example is a pronounced hot spot where the congestion in a single routing node's exit link determines when the network is saturated.

More generally, whenever some local congestion cannot be absorbed and balanced over the whole network, it will dominate the network at high load. If different configurations still have the same (or similar) bottleneck channel or channels (figures 22), the zero-load predictive model holds. If they have a different bottleneck channel, as may happen with deterministic routing algorithms, the average distance does not contain sufficient information to predict relative performance under load.

Thus, it needs to be emphasised, that the predictive power of the average distance model relies on the load distribution capability of adaptive routing. Our experiments have shown that it is less suitable for deterministic routing because in such networks individual links may constitute bottlenecks determining the limit of the network's load, even though the network as a whole has abundant spare capacity. The average distance model is a global property and it averages out local imbalances, thus mirroring closely the load distribution of adaptive routing. We have validated the model only for deflection routing, which, it can be argued, has a perfect load distribution capability. We hypothesize that the model is well suited for other adaptive routing algorithms to the extent that they have good load balancing capabilities; to confirm this hypothesis is future work.

Hence, it is ironic but understandable, that deflection routing together with other adaptive routing algorithms defies all attempts to formulate an accurate analytic delay model but finds in the average distance model a very good predictor of relative performance.

8. Conclusion

Delay models for NoCs with adaptive routing that can accommodate a range of spatio-temporal traffic patterns and topologies do not exist, due to the inherent complexity in capturing the effect of packet interaction across time and space. However we have have shown that a static, relative metric that does not consider congestion is able to predict with remarkable fidelity whether a network will exhibit better or worse performance than another, even under heavy loading and bursty traffic. This metric, the zero-load average distance, is a good predictor of the relative performance of NoCs with

adaptive routing because it is a global property that captures the essence of the load balancing capability of a network.

References

References

[1] A. E. Kiasari, A. Jantsch, Z. Lu, Mathematical formalisms for performance evaluation of networks-on-chip, ACM Computing Surveys.

[2] N. e. a. Audsley, Applying new scheduling theory to static priority preemptive scheduling, Software Engineering Journal 8 (5) (1993) 284–292.

[3] Z. L. Y. Qian, W. Dou, Analysis of worst-case delay bounds for on-chip packet-switching networks, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 29 (5) (2010) 802 –815. doi:10.1109/TCAD.2010.2043572.

[4] M. B. et al., Dataflow analysis for real-time embedded multiprocessor system design, in: Dynamic and Robust Streaming in and between Connected Consumer-Electronic Devices, Springer, 2005, pp. 81–108.

[5] P. Bogdan, R. Marculescu, Non-stationary traffic analysis and its implications on multicore platform design, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 30 (4) (2011) 508–519. doi:10.1109/TCAD.2011.2111270.

[6] J. Duato, P. López, Performance evaluation of adaptive routing algorithms for k-ary n-cubes, in: by Kevin Bolding, L. Snyder (Eds.), Proceedings of the First International Workshop on Parallel Computer Routing and Communication, Springer, 1994, pp. 45–59.

[7] J. Hu, R. Marculescu, Dyad: smart routing for networks-on-chip, in: Proceedings of the 41st annual Design Automation Conference, DAC, 2004, pp. 260–263.

[8] J. K. et al, A low latency router supporting adaptivity for on-chip interconnects, in: Proceedings of the 42nd Design Automation Conference, 2005, pp. 559–564.

[9] E. N. et al, Load distribution with the proximity congestion awareness in a network on chip, in: Proceedings of the Design Automation and Test Europe (DATE), 2003, pp. 1126–1127.

[10] L. P. L. Shang, A. Kumar, N. K. Jha., Thermal modeling, characterization and management of on-chip networks, in: Proceedings of the 37th MICRO, 2004.

[11] C. F. et al., Addressing transient and permanent faults in NoC with efficient fault-tolerant deflection router, IEEE Transactions on Very Large Scale Integration Systems (TVLSI) 21 (6) (2013) 1053–1066.

[12] P.-A. Tsai, Y.-H. Kuo, E.-J. Chang, H.-K. Hsin, A.-Y. Wu, Hybrid path-diversity-aware adaptive routing with latency prediction model in network-on-chip systems, in: VLSI Design, Automation, and Test (VLSI-DAT), 2013 International Symposium on, 2013, pp. 1–4. doi:10.1109/VLDI-DAT.2013.6533884.

[13] W. J. Dally, B. P. Towles, Principles and practices of interconnection networks, Elsevier, 2004.

[14] Z. Qian, D.-C. Juan, P. Bogdan, C.-Y. Tsui, D. Marculescu, R. Marculescu, A comprehensive and accurate latency model for network-on-chip performance analysis, in: Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific, 2014, pp. 323–328. doi:10.1109/ASPDAC.2014.6742910.

[15] S. Foroutan, Y. Thonnart, R. Hersemeule, A. Jerraya, An analytical method for evaluating network-on-chip performance, in: Proceedings of the Conference on Design, Automation and Test in Europe, DATE '10, European Design and Automation Association, 3001 Leuven, Belgium, Belgium, 2010, pp. 1629–1632.
URL http://dl.acm.org/citation.cfm?id=1870926.1871319

[16] U. Ogras, P. Bogdan, R. Marculescu, An analytical approach for network-on-chip performance analysis, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 29 (12) (2010) 2001–2013. doi:10.1109/TCAD.2010.2061613.

[17] C. D. Y. Boura, T. Jacob, A performance model for adaptive routing in hypercubes, in: Proceedings of the International Workshop on Parallel Processing, 1994, pp. 11–16.

[18] M. Ould-Khaoua, An analytical model of Duato's adaptive routing algorithm, IEEE Transactions on Computers 48 (12) (1999) 1–8.

[19] J. Duato, A new theory of deadlock-free adaptive routing in wormhole routing systems, IEEE Transactions on Parallel and Distributed Systems 4 (12) (1994) 1320–1331.

[20] H. Sarbazi-Azad, M. Ould-Khaoua, L. Mackenzie, Performance analysis of k-ary n-cubes with fully adaptive routing, in: Parallel and Distributed Systems, 2000. Proceedings. Seventh International Conference on, 2000, pp. 249–255. doi:10.1109/ICPADS.2000.857705.

[21] A. Khonsari, M. Ould-Khaoua, J. Ferguson, A general analytical model of adaptive wormhole routing in k-ary n-cube interconnection networks, SIMULATION SERIES 35 (2003) 547–554.

[22] G. M. et al, Performance modelling of adaptive routing in hypercubic networks under non-uniform and batch arrival traffic, in: 32nd IEEE Conference on Local Computer Networks, 2007, pp. 583–590.

[23] E. Kakoulli, V. Soteriou, T. Theocharides, Intelligent hotspot prediction for network-on-chip-based multicore systems, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 31 (3) (2012) 418–431. doi:10.1109/TCAD.2011.2170568.

[24] J. H. Bahn, N. Bagherzadeh, A generic traffic model for on-chip interconnection networks, in: NoCArc, First International Workshop on Network on Chip Architectures, 2008.

[25] M. W. et al., Data mining meets performance evaluation: Fast algorithms for modeling bursty traffic, in: ICDE, 2002.
URL citeseer.ist.psu.edu/article/wang01data.html

[26] R. M. Girish Varatkar, On-chip traffic modeling and synthesis for MPEG-2 video applications, IEEE Trans. on VLSI Syst 12 (1) (2004) 108–119.

[27] J. Duato, S. Yalamanchili, L. M. Ni, Interconnection networks: An engineering approach, Morgan Kaufmann, 2003.

[28] N. Nikitin, J. de San Pedro, J. Cortadella, Architectural exploration of large-scale hierarchical chip multiprocessors, Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 32 (10) (2013) 1569–1582. doi:10.1109/TCAD.2013.2272539.

[29] A. Agarwal, Limits on interconnection network performance, IEEE Transactions on Parallel and Distributed Systems 4 (6) (1991) 613–624.

[30] H. Liu, W. Lin, Y. Song, An efficient processor partitioning and thread mapping strategy for mesh-connected multiprocessor systems, in: Proc. ACM symposium on Applied computing, 1997.

[31] M. G. et al., Optimal network architectures for minimizing average distance in k-ary n-dimensional mesh networks, in: Proceedings of the Networks on Chip Symposium (NoCS), Pittsburgh, Pennsylvania, USA, 2011.

[32] A. Y. W. et al., Scalability of network-on-chip communication architecture for 3-d meshes, in: Proceedings of the International Symposium on Networks-on-Chip, San Diego, CA, 2009.

[33] T. Moscibroda, O. Mutlu, A case for bufferless routing in on-chip networks, in: Proceedings of the 36th Annual International Symposium on Computer Architecture, ISCA '09, ACM, New York, NY, USA, 2009, pp. 196–207. doi:10.1145/1555754.1555781.
URL http://doi.acm.org/10.1145/1555754.1555781

[34] C.-K. Hsu, K.-L. Tsai, J.-F. Jheng, S.-J. Ruan, C.-A. Shen, A low power detection routing method for bufferless noc, in: Quality Electronic Design (ISQED), 2013 14th International Symposium on, 2013, pp. 364–367. doi:10.1109/ISQED.2013.6523636.

[35] N. Zhang, H. Gu, Y. Yang, D. Fan, Qbnoc: Qos-aware bufferless noc architecture, Microelectronics Journal 45 (6) (2014) 751 – 758. doi:http://dx.doi.org/10.1016/j.mejo.2014.04.015.
URL http://www.sciencedirect.com/science/article/pii/S0026269214001050

[36] A. Y. W. et al., A scalable multi-dimensional NoC simulation model for diverse spatio-temporal traffic pattern, in: Proceedings of the 3D Systems Integration Conference (3DIC), San Francisco, California, USA, 2013.

[37] J. S. S. T. Association, et al., Jedec standard: Wide i/o single data rate specification (2011).

[38] P. Vivet, 3D integrated circuits: A memory-to-logic WideIO example, in: Design Impacts and 3D CAD Design Perspectives, HIPEAC RAPIDO WorkShop, 2022.

Biography

Awet Yemane Weldezion is a researcher in Electronic Systems Design at KTH - Royal Institute of Technology, Stockholm, Sweden. He received BSc(2000) in Electrical and Computer Engineering from Addis Ababa University, MSc(2006) in SoC design from KTH, MBA (2012) in Innovation and Growth from University of Turku - Finland. Since 2008, he is pursuing Ph.D. studies in Electronic Systems Design at KTH in areas of 3D-NoC.

Matt Grange received his MEng and PhD degrees in Electronic Systems Design from Lancaster University in the UK in 2007 and 2011 respectively. His PhD thesis focused on high-speed digital circuit applications and physical modeling for 3-D ICs. He currently works in the Calibre division of Mentor Graphics in Wilsonville, Oregon. His main interests are IC verification, thermal validation, RTL simulation and synthesis, place and route, interconnect modeling, NoCs, and emerging technologies.

Axel Jantsch (M97) received the Dipl.Ing. and Dr.Tech. degrees from the Technical University of Vienna, Vienna, Austria, in 1988 and 1992, respectively. He has been a Full Professor of electronic system design with the Royal Institute of Technology, Stockholm, Sweden, since December 2002. Currently he is a chair professor at Vienna University of Technology (TU Wien). His research interests include VLSI design and synthesis, system-level specification, modeling and validation, HW/SW co-design and co-syntheses, reconfigurable computing, and networks-on-chip.

Hannu Tenhunen received his MSc('82) from Helsinki University of Technology, Finland and PhD ('85) from Cornell University, Ithaca, NY, USA. Since 1992, he is chair professor at the Royal Institute of Technology. He was one of the originators of the interconnect centric design, globally asynchronous locally synchronous, and network-on-chip (NoC) paradigms. He has supervised over 70 M.Sc. thesis, 39 doctoral thesis, and 8 post-doc and published over 700 reviewed publications. During the last 20 years he has been actively involved in high technology policies, technology impact studies, innovations and changing the educational system.

Dinesh Pamunuwa (M04-SM09) received the B.Sc. degree (with honors) in EE Eng'g from the University of Peradeniya, Sri Lanka in 1997, and the Ph.D. degree in Electronic System Design from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2003. He was a Senior Lecturer (2004-2010) at Lancaster University and since 2011 a Reader in Microelectronics at University of Bristol. He has authored and coauthored over 60 international peer-reviewed articles in areas ranging from interconnect design and signal integrity issues, to methodologies and architectures for electronic system design and networks-on-chip, to nanoelectronics and nano-electro-mechanical (NEM) relay based circuit design.